

Amendments to the Claims

1 Claim 1 (currently amended): A computer program product for providing fast and efficient  
2 address lookup for an address comprised of a plurality of address components and wherein each  
3 address component is deemed to be more significant than its next-sequential neighboring address  
4 component, the computer program product embodied on one or more computer-readable media  
5 and comprising:

6 computer-readable program code means for creating a plurality of arrays comprising an  
7 array for each of the address components, wherein each array comprises a plurality of entries  
8 which are indexed using values of the address component for which the array was created, further  
9 comprising:

10 computer-readable program code means for obtaining a particular address value to  
11 be represented in the plurality of arrays;

12 computer-readable program code means for obtaining a bit mask associated with  
13 the particular address value;

14 computer-readable program code means for indexing into a highest-order one of  
15 the arrays using a most-significant component of the particular address value as an index element;

16 computer-readable program code means for setting a flag associated with the index  
17 element to on if the bit mask indicates that the next-sequential neighboring address component is  
18 considered significant, and for setting the flag to off otherwise; and

19 computer-readable program code means for repeating the indexing and setting,  
20 using the next-highest-order one of the arrays and the next-most-significant component of the  
21 particular address value, while the bit mask indicates that the next-sequential neighboring address

Serial No. 09/680,791

-6-

Docket RSW9-2000-0050-US1

22 component is considered significant, and for (1) storing information associated with the particular  
23 address value in a storage or memory location and (2) setting a pointer field associated with the  
24 index element to point to the storage or memory location, otherwise; and

25 computer-readable program code means for retrieving the stored information associated  
26 with a selected address value from the plurality of arrays, further comprising:

27 computer-readable program code means for obtaining the selected address value;

28 computer-readable program code means for obtaining a selected bit mask  
29 associated with the selected address value;

30 computer-readable program code means for indexing into the highest-order one of  
31 the arrays using the most-significant component of the selected address value as the index  
32 element; and

33 computer-readable program code means for determining that no result is available  
34 if the index element has no stored information, and for continuing otherwise, wherein the  
35 continuing further comprises:

36 computer-readable program code means for checking the flag associated  
37 with the index element; and

38 computer-readable program code means for returning the stored  
39 information from the storage or memory location pointed to by the pointer field when the flag is  
40 set off or for repeating the indexing and determining, for the next-highest-order one of the arrays  
41 and the next-most-significant component of the selected address value, when the flag is set on.

1 Claim 2 (original): The computer program product according to Claim 1, wherein the computer-

2 readable program code means for repeating further comprises computer-readable program code  
3 means for setting a use count associated with the storage or memory location to a number which  
4 represents a count of the array entries which point to this storage or memory location when the  
5 next-sequential neighboring address component is not considered significant.

1 Claim 3 (original): The computer program product according to Claim 2, wherein the stored  
2 information in the memory or storage location comprises an associated bit mask and wherein the  
3 computer-readable program code means for retrieving further comprises computer-readable  
4 program code means for resolving a collision, further comprising:

5 computer-readable program code means for comparing the selected address value to each  
6 bit mask associated with the stored information from multiple storage or memory locations,  
7 yielding a plurality of bit mask results; and

8 computer-readable program code means for selecting a collision result using that one of  
9 the bit mask results which both (1) matches the selected address value according to the selected  
10 bit mask and (2) has the longest associated bit mask.

1 Claim 4 (original): The computer program product according to Claim 1, wherein the address is  
2 an Internet Protocol (IP) address.

1 Claim 5 (original): The computer program product according to Claim 4, wherein the IP address  
2 is an IP version 4 address and wherein there are 4 components in each IP version 4 address and  
3 thus 4 arrays.

Serial No. 09/680,791

-8-

Docket RSW9-2000-0050-US1

1 Claim 6 (original): The computer program product according to Claim 4, wherein the IP address  
2 is an IP version 6 address and wherein there are 16 address components in each IP version 6  
3 address and thus 16 arrays.

1 Claim 7 (currently amended): A system for providing fast and efficient address lookup for an  
2 address comprised of a plurality of address components and wherein each address component is  
3 deemed to be more significant than its next-sequential neighboring address component, the system  
4 comprising:

5 means for creating a plurality of arrays comprising an array for each of the address  
6 components, wherein each array comprises a plurality of entries which are indexed using values of  
7 the address component for which the array was created, further comprising:

8 means for obtaining a particular address value to be represented in the plurality of  
9 arrays;

10 means for obtaining a bit mask associated with the particular address value;

11 means for indexing into a highest-order one of the arrays using a most-significant  
12 component of the particular address value as an index element;

13 means for setting a flag associated with the index element to on if the bit mask  
14 indicates that the next-sequential neighboring address component is considered significant, and for  
15 setting the flag to off otherwise; and

16 means for repeating the indexing and setting using the next-highest-order one of  
17 the arrays and the next-most-significant component of the particular address value, while the bit

mask indicates that the next-sequential neighboring address component is considered significant, and for (1) storing information associated with the particular address value in a storage or memory location and (2) setting a pointer field associated with the index element to point to the storage or memory location, otherwise; and

means for retrieving the stored information associated with a selected address value from the plurality of arrays, further comprising:

means for obtaining the selected address value;

means for obtaining a selected bit mask associated with the selected address value;

means for indexing into the highest-order one of the arrays using the most-significant component of the selected address value as the index element; and

means for determining that no result is available if the index element has no stored information, and for continuing otherwise, wherein the continuing further comprises:

means for checking the flag associated with the index element; and

means for returning the stored information from the storage or memory location pointed to by the pointer field when the flag is set off or for repeating the indexing and determining, for the next-highest-order one of the arrays and the next-most-significant component of the selected address value, when the flag is set on.

Claim 8 (original): The system according to Claim 7, wherein the means for repeating further comprises means for setting a use count associated with the storage or memory location to a number which represents a count of the array entries which point to this storage or memory location when the next-sequential neighboring address component is not considered significant.

Serial No. 09/680,791

-10-

Docket RSW9-2000-0050-US1

1 Claim 9 (original): The system according to Claim 8, wherein the stored information in the  
2 memory or storage location comprises an associated bit mask and wherein the means for  
3 retrieving further comprises means for resolving a collision, further comprising:

4 means for comparing the selected address value to each bit mask associated with the  
5 stored information from multiple storage or memory locations, yielding a plurality of bit mask  
6 results; and

7 means for selecting a collision result using that one of the bit mask results which both (1)  
8 matches the selected address value according to the selected bit mask and (2) has the longest  
9 associated bit mask.

1 Claim 10 (original): The system according to Claim 7, wherein the address is an Internet Protocol  
2 (IP) address.

1 Claim 11 (original): The system according to Claim 10, wherein the IP address is an IP version 4  
2 address and wherein there are 4 components in each IP version 4 address and thus 4 arrays.

1 Claim 12 (original): The system according to Claim 10, wherein the IP address is an IP version 6  
2 address and wherein there are 16 address components in each IP version 6 address and thus 16  
3 arrays.

1 Claim 13 (currently amended): A method for providing fast and efficient address lookup for an

2 address comprised of a plurality of address components and wherein each address component is  
3 deemed to be more significant than its next-sequential neighboring address component, the  
4 method comprising the steps of:

5 creating a plurality of arrays comprising an array for each of the address components,  
6 wherein each array comprises a plurality of entries which are indexed using values of the address  
7 component for which the array was created, further comprising the steps of:

8 obtaining a particular address value to be represented in the plurality of arrays;

9 obtaining a bit mask associated with the particular address value;

10 indexing into a highest-order one of the arrays using a most-significant component  
11 of the particular address value as an index element;

12 setting a flag associated with the index element to on if the bit mask indicates that  
13 the next-sequential neighboring address component is considered significant, and setting the flag  
14 to off otherwise; and

15 repeating the indexing and setting, using the next-highest-order one of the arrays  
16 and the next-most-significant component of the particular address value, while the bit mask  
17 indicates that the next-sequential neighboring address component is considered significant, and (1)  
18 storing information associated with the particular address value in a storage or memory location  
19 and (2) setting a pointer field associated with the index element to point to the storage or memory  
20 location, otherwise; and

21 retrieving the stored information associated with a selected address value from the  
22 plurality of arrays, further comprising the steps of:

23 obtaining the selected address value;

24 obtaining a selected bit mask associated with the selected address value;  
25 indexing into the highest-order one of the arrays using the most-significant  
26 component of the selected address value as the index element; and  
27 determining that no result is available if the index element has no stored  
28 information, and continuing otherwise, wherein the continuing further comprises the steps of:  
29 checking the flag associated with the index element; and  
30 returning the stored information from the storage or memory location  
31 pointed to by the pointer field when the flag is set off or repeating the indexing and determining,  
32 for the next-highest-order one of the arrays and the next-most-significant component of the  
33 selected address value, when the flag is set on.

1 Claim 14 (original): The method according to Claim 13, wherein the repeating step further  
2 comprises the step of setting a use count associated with the storage or memory location to a  
3 number which represents a count of the array entries which point to this storage or memory  
4 location when the next-sequential neighboring address component is not considered significant.

1 Claim 15 (original): The method according to Claim 14, wherein the stored information in the  
2 memory or storage location comprises an associated bit mask and wherein the retrieving step  
3 further comprises resolving a collision, further comprising the steps of:  
4 comparing the selected address value to each bit mask associated with the stored  
5 information from multiple storage or memory locations, yielding a plurality of bit mask results;  
6 and



7 selecting a collision result using that one of the bit mask results which both (1) matches  
8 the selected address value according to the selected bit mask and (2) has the longest associated bit  
9 mask.

1 Claim 16 (original): The method according to Claim 13, wherein the address is an Internet  
2 Protocol (IP) address.

1 Claim 17 (original): The method according to Claim 16, wherein the IP address is an IP version 4  
2 address and wherein there are 4 components in each IP version 4 address and thus 4 arrays.

1 Claim 18 (original): The method according to Claim 16, wherein the IP address is an IP version 6  
2 address and wherein there are 16 address components in each IP version 6 address and thus 16  
3 arrays.

1 Claim 19 (currently amended): A method for providing fast and efficient address lookup for an  
2 address comprised of a plurality of address components, the method comprising the steps of:  
3 creating a plurality of arrays comprising an array for each of the address components,  
4 wherein each array comprises a plurality of entries which are indexed using values of the address  
5 component for which the array was created; and  
6 storing entries and information for each address to be subsequently looked up, further  
7 comprising the steps of:  
8 creating an entry for a particular address using the plurality of arrays; and

9 storing information associated with the particular address value in a storage or  
10 memory location associated with a last significant component of the entry, wherein the last  
11 significant component is determined by a bit mask associated with the particular address; and  
12 ~~retrieving the stored information associated with a selected address value from the~~  
13 ~~plurality of arrays.~~

1 Claim 20 (new): The computer program product according to Claim 1, wherein the computer-  
2 readable program code means for retrieving is performed by a plurality of distinct processors,  
3 each operating on different ones of the components of the selected address value.

1 Claim 21 (new): The computer program product according to Claim 1, wherein the stored  
2 information for each of the addresses comprises routing table information associated with a route  
3 to that address.

1 Claim 22 (new): The system according to Claim 7, wherein the means for retrieving is performed  
2 by a plurality of distinct processors, each operating on different ones of the components of the  
3 selected address value.

1 Claim 23 (new): The system according to Claim 7, wherein the stored information for each of the  
2 addresses comprises routing table information associated with a route to that address.

1 Claim 24 (new): The method according to Claim 13, wherein the retrieving step is performed by

2 a plurality of distinct processors, each operating on different ones of the components of the  
3 selected address value.

1 Claim 25 (new): The method according to Claim 13, wherein the stored information for each of  
2 the addresses comprises routing table information associated with a route to that address.

1 Claim 26 (new): The method according to Claim 19, further comprising the step of subsequently  
2 looking up a selected address value by retrieving the stored information associated with the  
3 selected address value from the plurality of arrays.